



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/828,439	04/06/2001	Humberto A. Sanchez II	10006054-1	6175

7590 07/08/2004

HEWLETT-PACKARD COMPANY  
Intellectual Property Administration  
P.O. Box 272400  
Fort Collins, CO 80527-2400

EXAMINER

BROSS, EDWARD J

ART UNIT	PAPER NUMBER
2126	

DATE MAILED: 07/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>
	09/828,439	SANCHEZ, HUMBERTO A. 
	<b>Examiner</b>	<b>Art Unit</b>
	Edward Bross	2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### **Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

1)  Responsive to communication(s) filed on 06 April 2001.

2a)  This action is **FINAL**.                            2b)  This action is non-final.

3)  Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

4)  Claim(s) 1-20 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
5)  Claim(s) \_\_\_\_\_ is/are allowed.  
6)  Claim(s) 1-20 is/are rejected.  
7)  Claim(s) \_\_\_\_\_ is/are objected to.  
8)  Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

9)  The specification is objected to by the Examiner.

10)  The drawing(s) filed on \_\_\_\_\_ is/are: a)  accepted or b)  objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)  The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)  Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a)  All b)  Some \* c)  None of:  
1.  Certified copies of the priority documents have been received.  
2.  Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3.  Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1)  Notice of References Cited (PTO-892)  
2)  Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3)  Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 4/6/01.

4)  Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_ .  
5)  Notice of Informal Patent Application (PTO-152)  
6)  Other: \_\_\_\_\_

### **DETAILED ACTION**

1. Claims 1-20 are pending in this application.

#### ***Claim Rejections - 35 USC § 112***

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 1-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claims are indefinite for the following reasons:

- a. Claims 1, 13 and 19 – it is uncertain what is meant by “and methods by” (i.e., is it a method or a systems claim).
- b. Claims 16 and 17 – it is not clear what is meant by “method data” p. 21 lines 27, 29, 31-32 and 34.

#### ***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cheng et al. (6,182,155) in view of The Java Tutorial (JT).

6. As to claims 1, 13, 19 and 20 Cheng discloses the invention substantially as claimed including:

a Java Virtual Machine (implied by the presence of Java objects, col. 6, lines 39-40, Fig. 4);

a C++ environment, wherein a call across a boundary is required to access the Java Virtual Machine from the C++ environment (Fig. 4, CASE B), the C++ environment comprising:

a base proxy object, comprising one or more functions that encapsulate one or more calls necessary to call a Java method in the Java Virtual Machine (col. 5, lines 50-57).

7. Cheng does not disclose the C++ is a graphical user interface,

a Java Native Interface (“JNI”) boundary;

the boundary is a JNI boundary,

the calls are callbacks from a graphical user interface.

8. JT discloses a Java Native Interface Boundary and callbacks (p. 1 “Calling Java Methods” lines 2-3). JT does not disclose a C++ graphical user interface. However, C++ graphical user interfaces are well known in the art.

9. It would have been obvious to one of ordinary skill in the art at the time of the invention to use a JNI and callbacks of JT in the system of Cheng with a C++ graphical user interface in order to leverage the tools available for this standardized cross-language API.

10. As to claims 2 and 15, Cheng discloses the Java Virtual Machine comprises:
  - a Java object (col. 6, lines 39-40), comprising:
    - an attribute (common to Java objects); and
    - one or more methods that are executed to enter, retrieve or modify the attribute (col. 6, lines 35-40); and
  - wherein the base proxy object makes the one or more calls across the boundary to call the one or more methods of the Java object based on the one or more method requests (col. 6, lines 33-48).
11. Cheng does not disclose that the boundary is a JNI boundary. However, JT discloses a JNI boundary as in claim 1 above.
12. As to claim 3, Cheng discloses the C++ environment further comprises:
  - a C++ proxy object that proxies the Java object, the C++ proxy object comprising:
    - one or more methods that correspond to the one or more methods of the Java object and that call one or more functions of the base proxy object when executed, wherein the one or more methods of the C++ proxy object are executed in response to one or more method requests (col. 5 line 580col. 6 line 8).
13. As to claim 4, Cheng and JT do not explicitly discloses the C++ graphical user interface executes for a finite length of time and the C++ proxy object and the Java object exist in the C++ environment and the Java virtual machine during the C++ graphical user interface execution; However, given a C++ GUI used in conjunction with the system of Cheng and JT, it is implied

by necessity that the GUI would execute for a finite period of time and objects would exist during this time.

14. As to claim 5, Cheng and JT do not disclose the Java object is an instance of an instantiated Java class and the C++ proxy object is created as a result of the instantiation of the Java class. However, creation of a proxy object upon instantiation of a class is well known in the art (such as instantiation of CORBA objects).

15. It would have been obvious to one of ordinary skill in the art at the time of the invention to create the C++ proxy object upon instantiation of the Java class in the system of Cheng and JT in order to minimize system resources used until needed.

16. As to claims 6 and 18, Cheng does not disclose the C++ proxy object includes instance data that identifies the Java object and locates the Java object in the Java virtual machine and wherein the instance data is passed from the Java virtual machine to the C++ proxy object when the C++ proxy object is created.

17. JT discloses the C++ proxy object includes instance data that identifies the Java object and locates the Java object in the Java virtual machine and wherein the instance data is passed from the Java virtual machine to the C++ proxy object when the C++ proxy object is created (p. 1 "Calling a Java Method from Native Code" code excerpt lines 1-4).

18. As to claims 7 and 16, Cheng does not disclose the C++ proxy object includes one or more method names that name the one or more methods of the Java object and wherein the C++

proxy object passes the one or more method names to the base proxy object when calling the one or more functions of the base proxy object.

19. JT discloses the C++ proxy object includes one or more method names that name the one or more methods of the Java object and wherein the C++ proxy object passes the one or more method names to the base proxy object when calling the one or more functions of the base proxy object (p. 1 “Calling a Java Method from Native Code” code excerpt line 5).

20. As to claim 8, Cheng does not disclose one or more method 1Ds identify the one or more methods of the Java object and the base proxy object retrieves the one or more method IDs using the one or more method names provided by the C++ proxy object.

21. JT discloses one or more method 1Ds identify the one or more methods of the Java object and the base proxy object retrieves the one or more method IDs using the one or more method names provided by the C++ proxy object (p. 1 “Calling a Java Method from Native Code” code excerpt line 5).

22. As to claims 9 and 17, Cheng does not disclose the base proxy object passes the one or more method IDs to the Java virtual machine when making the one or more JNI API calls across the JNI boundary to call the one or more methods of the Java object.

23. JT discloses the base proxy object passes the one or more method 1Ds to the Java virtual machine when making the one or more JNI API calls across the JNI boundary to call the one or more methods of the Java object (p. 1 “Calling a Java Method from Native Code” code excerpt line 10).

24. As to claim 10, Cheng and JT do not disclose the base proxy object caches the one or more method IDs in a C++ hash table that is accessible by the C++ proxy objects and the base proxy object. However, the use of hash tables to cache values is well known in the art.

25. It would have been obvious to one of ordinary skill in the art at the time of the invention to use a hash table to cache the method IDs in the system of Cheng and JT in order to reduce latency in calling methods across the boundary.

26. As to claim 11, Cheng and JT do not disclose the Java object is one of the following: a user object, for adding or modifying a user; a node object, for adding or modifying a node; a node group object, for adding or modifying a node group; a tool object, for adding or modifying a tool; and a role object, for adding or modifying a role. However, these types of objects are well known in the art (such as a object representing a node in a tree structure).

27. It would have been obvious to one of ordinary skill in the art at the time of the invention to use a node object with the system of Cheng and JT in order to implement tree data structures that are accessible from both C++ and Java environments.

28. As to claim 12, Cheng does not discloses the base proxy object further comprises a mapping mechanism for mapping Java data types to C++ data types.

29. JT discloses a mapping mechanism for mapping Java data types to C++ data types (pp. 6-7 "Mapping Between Java and Native Types").

30. As to claim 14, Cheng does not disclose if the executed Java method is a get method, returning a pointer to C++ data.

31. JT discloses if the executed Java method is a get method, returning a pointer to C++ data (return type of jobject, p. 6 "Java Object Types").

32. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Edward Bross whose telephone number is 703-305-8754. The examiner can normally be reached on Mon-Fri 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 703-305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

EB

  
MENG-AI T. AN  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100